Modeling QoE in Dependable Tele-Immersive Applications: A Case Study of World Opera

Modeling QoE in Dependable Tele-immersive Applications: A Case Study of World Opera

Narasimha Raghavan Veeraragavan, Leonardo Montecchi, Nicola Nostro, Roman Vitenberg, Hein Meling, and Andrea Bondavalli

***Abstract*—With the advent of recent technological advances, more demanding tele-immersive applications have started to emerge. In the World Opera application, artists from different opera houses across the globe can participate in a single united performance, and interact almost as if they were co-located.**

One of the main design challenges in this application domain is to assess to what extent the inevitable failures of some of the numerous and complex hardware, software, and network components affect the quality of experience for the user. This challenge cannot be addressed by traditional system-centric methods for dependability evaluation, which do not take personalized user perspective into account when considering meaningful and acceptable degradation of services.

In this paper, we propose a novel method to assess the quality of experience in presence of failures, based on a new metric called *perceived reliability*. The method takes the human perspective into account and allows considering factors such as human perception of video and audio, characteristics of the audience, as well as performance elements and artistic content. This method can help system designers and engineers compare architectural variants and determine the dependability budget.

We show the feasibility of our method by applying it to a World Opera performance. To this end, we construct a SAN-based model and run simulations in the Möbius framework. The obtained results provide useful guidelines for system engineers towards improving the quality of experience of World Opera performances despite the presence of failures.

***Index Terms*—Reliability analysis, Quality of Experience, Tele-immersive Applications, World Opera**

I. Introduction

Tele-immersive (TI) applications, such as Massive Online Multiplayer Games [1] and video conferencing systems have become commonplace among Internet users. These applications share a number of demanding traits, such as their real-time and interactive nature, which imposes stringent requirements on latency and synchronization. However, these classical TI applications are limited in terms of their interaction complexity, types and number of streams, and number of participants at each location. Recent technological advances have enabled the design and implementation of even more demanding TI applications, in which the bandwidth requirement vastly surpasses that of classical multimedia applications.

N. Raghavan Veeraragavan and R. Vitenberg are with the University of Oslo, Norway; {raghavan,romanvi}@ifi.uio.no. L. Montecchi, N. Nostro, and A. Bondavalli are with the University of Firenze, Italy; {lmontecchi,nnostro,bondavalli}@unifi.it. H. Meling is with the University of Stavanger, Norway; hein.meling@uis.no.

This work was partially supported by the Tidal News project under grant no. 201406 from the Research Council of Norway, the TENACE PRIN Project (no. 20103P34XC) funded by the Italian Ministry of Education, University and Research, and by the DEVASSES project, funded by the EU 7th Framework Programme under grant agreement PIRSES-GA-2013-612569.

Fig. 1: World Opera rehearsal

One such application is World Opera (WO), an application envisioned by the WO artistic consortium [2].

The WO consortium and its partners are engaged in conducting distributed, real-time, live opera performances across several world renowned opera houses. Each opera house represents a real-world stage with its own musicians, singers, dancers, and actors. Interaction between the artists is orchestrated by a single conductor present at a single selected stage. The artists on all participating stages can perform together, and interact almost as if they were co-located. Such a combined performance is projected as video on display devices, and shown to the audience at both local and remote opera houses.

Figure 1 shows a WO rehearsal at the Music Conservatory of the University of Tromsø, where a singer coordinates with an actor and a pianist, placed at three different locations. More details about the rehearsals are available in [2]. A key observation derived from early WO performances reveals that it is notoriously difficult to maintain a smooth technical operation for the entire duration of a performance, even after taking proper preparatory steps.

Towards addressing this challenge, we observe that, while it may be difficult to deliver a flawless performance in TI applications such as WO, these applications are amenable to meaningful graceful degradation. For example, the audience may, for a moderate amount of time, find it acceptable to hear the orchestra without seeing it. Secondly, we advocate that subjective factors, including the perception and other user characteristics, have a key role in correctly evaluating the reliability of multimedia applications. For example, consider a distributed dance performance, where the audience is more immersed in the video than the audio. A short transient failure affecting only the audio may be unperceivable by the audience, as explained by selective attention theory in psychology [3].

Furthermore, a completely failure-free execution is not necessarily required to accomplish a successful performance. For instance, the sound from the same group of artists is

typically captured by multiple microphones. While the failure of one microphone may somewhat affect the *quality of experience* (QoE), the audience may tolerate it, either during the entire performance, or just for a limited period of time. Therefore, the classical notion of reliability [4] (*continuous* delivery of correct service) is not an appropriate metric for evaluating TI applications, in which partial and intermittent failures may not necessarily impair the performance.

To this end, we propose a novel reliability metric that captures user-perceived QoE. This user-centric metric, called *perceived reliability*, allows brief unperceivable failures and formalizes multiple QoE degradation levels, which can be configured for a specific application or performance. Furthermore, different levels can be defined for individual users or user categories or even a combination of both for a given application. We show how QoE levels can be defined for the WO application based on the feedback from artists and stage engineers from recent WO experimentations.

In order to apply perceived reliability and evaluate QoE in presence of failures, we construct a model for evaluating distributed TI applications. The model is realized using Stochastic Activity Networks (SANs) [5], and built using a modular approach, defining "model templates" for different elements of the WO architecture, which can then be composed together in different ways to represent different system configurations and architectures. Coupled with the configurable definition of perceived reliability, this framework allows us to assess the metric in presence of different failures occurring in the system, and for a broad range of system configurations, thus facilitating decisions about the design and configuration for WO performances.

We evaluate both classic and perceived reliability for WO by running model simulations in the Möbius framework [6]. The values for most parameters used in the simulations are based on WO experiments or existing studies in psychology. For the other parameters, we conduct a sensitivity analysis by varying these parameters within a reasonable range of values.

In summary, we provide the following key contributions in this paper: i) we apply a novel approach to capture the concept of meaningful degradation in TI applications; ii) we define a new QoE-aware metric called perceived reliability to account for the subjective perception of different users in TI applications; iii) we design and implement a modeling framework to evaluate this metric for a broad range of configurations; and iv) we apply the framework to a WO performance, providing useful guidelines to stage engineers.

The remainder of this paper is organized as follows. We present TI applications in general and WO in particular in Sections II and III, respectively. We survey existing definitions for QoE and reliability in Section IV and explain why they are not adequate in our context. We describe our new approach for dependability modeling of QoE in TI and introduce the concept of perceived reliability in Section V, as a metric to evaluate QoE. It is shown in Section VI how this general approach can be instantiated for WO. The construction of the SAN model is presented in Section VII. We report on the experimental results in Section VIII. Finally, we describe the related work in Section IX and conclude in Section X.

II. Tele-immersive Applications

To realize the vision of a mixed-reality environment, with highly immersive and interactive communication among distributed participants, several research fields such as virtual reality, haptics, high-speed networking, computer vision, etc., must be integrated. The technology that enables this integration is known as tele-immersion, and tele-immersive (TI) applications implement that technology. Some example TI applications include: distributed musical performances [7], distributed dance performances [8], tele-immersive gaming [9].

A TI application can be viewed as a collection of distributed mixed-reality worlds, which represent a combination of real and virtual worlds. Real worlds are actual physical locations at which the TI participants reside. These locations are geographically distributed. A virtual world is a display of users from multiple remote real worlds. It is usually projected as video on display devices at real-world stages. Additionally, virtual-world stages can include imaginary elements, such as animated cartoon characters mimicking the behavior of artists at remote stages.

A TI application is composed of *components* and *streams* between these components. A component can for example be a microphone or a workstation. We distinguish between different components based on their function: *acquiring components*, acquire data from the real world; *processing components*, process acquired data; and *playback components*, reproduce multimedia data to the real world.

A stream is a basic element of a TI application contributing to the performance, e.g., audio or video streams. Each stream is captured by a set of acquiring components located in the same stage (e.g., an array of microphones), and then flows through a set of processing components, possibly located in different stages. Finally, it reaches playback components, which reproduce the stream to the participating users.

The activities at each real-world stage are divided into five phases: initialization, capturing, processing, streaming, and rendering. During the *initialization phase*, the stage operators agree on the set of components to be used for each stage. In the *capturing phase*, the corresponding components receive activation signals and start generating streams. There exist two principal stream types: video and audio. These generated streams collectively represent the real-world data.

In the *processing phase*, all generated streams are processed to remove noise. Additionally, video streams are compressed, timestamped, and processed using computer vision techniques for artistic reasons. The *streaming phase* is where the streams are multicast and received by the remote stages. In the *rendering phase*, the received streams are uncompressed and synchronized based on their timestamps, and finally rendered to the virtual world.

III. Case Study: The World Opera Application

With the WO application, the WO consortium aims to conduct real-time, live opera performances across several opera houses, each with its own set of artists. Interaction between the artists is orchestrated by a single *conductor* present at one of the stages, typically the stage where the most artists and technical

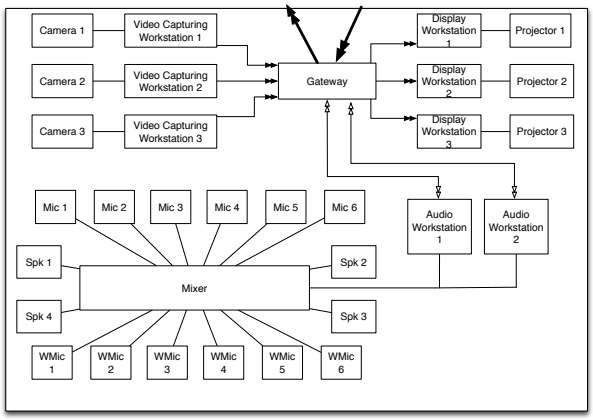


Fig. 2: System architecture of a World Opera stage.

components are located. In addition, each stage has a *director*, who is a technician with artistic knowledge, responsible for managing the technical components based on the artistic requirements. A typical WO setup would consist of 3–7 stages.

A. World Opera Architecture

In our analysis, we ignore the initialization phase since it is performed offline before the performance begins. Failures in this phase do not significantly impact the reliability of the online performance since the repair rate in this phase is high. That is, there is a high probability that failed components are replaced before leaving the initialization phase. We exclude the processing phase from our models in this paper because it is not part of the current WO deployment.

The architecture for a WO stage is shown in Figure 2. The capturing phase uses the following components: cameras, wired and wireless microphones (acquiring components), camera workstations, and a mixer (processing components). Cameras are used to capture the video streams portraying the artists from multiple view-points in the real world. Camera workstations are used to receive video streams and control the motion of the cameras during the performance. Several camera workstations send their video streams to the gateway. The microphones are organized into an array controlled by the mixer. The mixer is responsible for activation of microphones, adjusting the audio signals, and routing the audio stream. The audio streams are routed to multiple speakers via the mixer, and to the remote stages through the gateway via audio workstations.

The streaming phase involves a single processing component: the gateway, which is responsible for multicasting the video and audio streams to the remote stages through a dedicated high-speed connection of at least 10 Gbps. These streams are received at the remote stages and routed to the display and audio workstations.

The rendering phase has the following components: display and audio workstations, and a mixer (processing components), and projectors and speakers (playback components). The audio and video streams are received by the audio and display workstations, respectively. The audio workstations render the audio streams to the speakers through the mixer. The display workstations render the images on multiple projectors.

In order to cope with component failures, we assume that components have hot-spares. This design choice is reasonable, since repairing a component during a WO performance is too time consuming to be practical. Therefore, all the components are considered non-repairable during the performance: if a component has failed, its functionality can be restored by switching to an identical spare.

IV. Analysis of QoE and Dependability Concepts in Existing Tele-immersive Applications

In this section, we identify several key attributes of TI applications that are currently not covered by existing QoE and dependability metrics. In Section V, we show how we propose to incorporate these attributes in our modeling framework.

Research into QoE for TI is still in its infancy; most efforts have been geared towards understanding and addressing the functional requirements of TI applications. The effect of failures and fault-tolerant mechanisms on QoE has, to our knowledge, not been considered yet. However, this effect can be significant: for example, both the timeliness of a recovery mechanism and the number of backup components play an important role for QoE, as we show in Section VIII.

On the other hand, traditional methods and metrics for dependability evaluation mainly focus on system-centric aspects in the form of QoS specifications, and fail to capture the human-centric perspective of quality. As a concrete example, the human brain is unable to discern loss of video for a short period in the order of tens of milliseconds. This makes effective failure masking in TI easier compared to systems that demand uninterrupted service execution, which are typically studied by dependability research; this also implies that the classic reliability concept is not an appropriate indicator. The significance of the human perspective goes way beyond this simple example, as we explain in the remainder of this section.

A. State-of-the-art QoE Definitions

The QoE term has been gaining traction lately due to the shift away from system-centric towards human-centric evaluation when deploying products, applications, and services [10]. However, even though human-centric QoE is fundamental, no single unifying definition has been agreed upon. Below we summarize a number of existing definitions.

ITU-T [11] defines QoE as "The overall acceptability of an application or service, as perceived subjectively by the end-user." Accordingly, the QoE is assessed as a one-dimensional subjective metric called Mean Opinion Score (MOS), a Likert-scale [12] for subjective ranking of voice and video quality. Typically, a MOS score in the range 1 (bad) – 5 (excellent), represents a user's QoE. This definition has been criticized for its narrow interpretation [13], [14], [15].

In [16] the authors adopt the following definition: "QoE is a multi-dimensional construct of perceptions and behaviors of a user, which represent his/her emotional, cognitive, and behavioral responses, both subjective, and objective, while using a system." Most of the existing evaluations [17], [18], [16], [19] of TI follow variants of the above definition.

None of the existing definitions capture how QoE should be assessed in the presence of failures. Furthermore, the same TI may provide different levels of QoE to different users in the presence of failures, as we show in Section VIII. Hence, it is important to understand the QoE requirements for the various stakeholders involved in a TI application.

B. Diversity of QoE Requirements in TI Applications

There is a number of factors that contribute to the diversity of QoE requirements in TI applications:

Type of application: While video is the most critical element for a pantomime theater, audio plays the most important role for a musical performance.

Artistic content: Even in the same performance, the requirements may be different depending on the specific content. For example, a more rhythmic music requires lower latency compared to long tones, thereby making failures more difficult to mask, as confirmed in the WO experiments [19].

Participant characteristics: If participants are located at different distances from the speakers, they may have different audio QoE requirements, depending on the acoustics of the theater. Professional artists might have higher requirements compared to amateurs. Different QoE might be provided to different classes of users based on the business model. However, the most significant factor for the diversity of QoE requirements is the variety of participant roles.

QoE evaluations [17], [18], [16], [19] for TI typically adopt a coarse-grained classification with two user roles: users who participate in the performance (e.g., musicians) and users who watch it (e.g., the audience). We claim that in order to provide a realistic evaluation of a large-scale TI application under failures, the division should be more refined and based on participant activities and interaction patterns.

For example, consider a scenario where a singer, a pianist, and the audience are located at three different stages. In this scenario, the singer primarily relies on remote audio, and may still perceive an acceptable QoE even though the video of the remote pianist is missing due to some component failure. On the other hand, the audience may perceive a more significant degradation in QoE as they expect to see the video of the remote pianist as well.

The above example shows that, in presence of failures, QoE may differ from one user role to another, even during the same TI performance. Such a diversity poses a significant challenge for addressing dependability in TI applications. For a large-scale TI application with a large number of participants of different types and characteristics, there is a need to define QoE requirements for all participants involved and to satisfy such requirements, possibly providing redundancy and recovery mechanisms. The challenge is exacerbated by the fact that dependability modeling techniques have scalability limitations with respect to the number of technical components in the system as well as QoE requirements.

C. Lack of Human Perspective in Dependability Analysis

The existing dependability metrics such as reliability, availability, maintainability, and safety, and dependability mechanisms

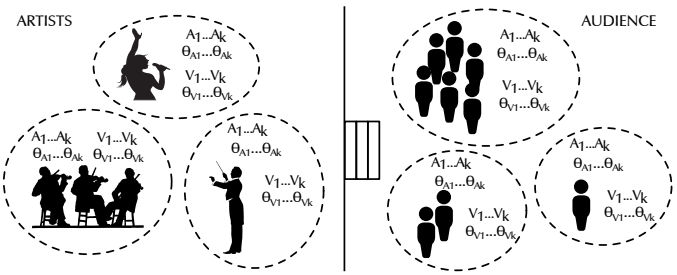


Fig. 3: The QoE levels $A_1 \dots A_k$ and $V_1 \dots V_k$, as well as the associated θ values, may differ across different users or groups.

are based on the system-centric evaluation (QoS). Unfortunately, the classical definition of reliability (*the ability to continuously provide correct service* [4]) is too restrictive for TI as it does not account for the actual perception of the user, who may not be able to discern an ephemeral quality degradation. This is due to two factors [3]: (i) the inability of the human brain to register changes below a certain duration, and (ii) the cognitive limitation of an individual/group to simultaneously focus their attention on multiple aspects of a performance, as documented by selective attention theory from psychology [3]. For example, in a distributed dance performance, where the audience is immersed in the video, the audience may not notice a small change in audio quality.

Moreover, providing a performance that is failure-free in the classical sense is notoriously difficult and unlikely [20]. Fortunately, the audience might be able to tolerate a mild failure even if the failure is perceived, if a higher quality level is restored within a reasonable amount of time.

However, as the human perception and expectations vary based on the user roles in a TI application, the dependability metric and its computation should take this diversification into account. Furthermore, different user roles focus on distinct elements of the artistic performance, which are supported by different sets of hardware and software functions/services/components. For all these reasons, the reliability of the same performance may appear different for different individuals or user roles.

V. Integrating QoE and Dependability

In this section, we present our approach for addressing dependability modeling of QoE in TI. To this end, we introduce the concepts of unperceivable failure, tolerable duration, QoE levels, and perceived reliability, and show how they work together in the context of dependability modeling.

An architecture for a TI application consists of a large number of hardware and software components distributed across different locations. These components collectively produce video and audio as explained in Section III. We observe that streams provide the right level of abstraction for defining QoE requirements: it is rendered streams, or lack thereof, that affect the perception of the users.

We define *unperceivable failure* as a failure of a stream that is shorter than the *perceptibility threshold* at which it can be perceived by the user. The duration of the perceptibility threshold is determined by the two factors identified in Section IV-C.

A performance with stream failures shorter than the threshold may still be perceived as perfect by the users.

Even when a stream failure is perceived, the resulting QoE will be degraded, yet it may be acceptable as explained in Section IV-C. Thus, a fundamental step for dependability modeling in TI is defining degraded but still *acceptable QoE levels*. While the concept of QoE levels is generic and important for all TI applications, the concrete definition in the context of a specific application depends on two major factors: the artistic content and diversity of user requirements due to user roles, physical location w.r.t. the stage, individual human characteristics such as the mood or level of artistic professionalism, etc. We provide a concretization of QoE levels for WO in Section VI.

Using the concepts of an unperceivable failure and QoE levels, we propose a subjective quality metric which we call *perceived reliability* as the central metric for evaluating the dependability of TI applications. Conceptually, perceived reliability with respect to a specific QoE level is the probability that the performance meets certain criteria with respect to this level. Similarly to classical reliability, perceived reliability is a probabilistic value in the interval $[0, 1]$, wheareas 1 means that the QoE expected by the user is always accomplished, and 0 means that the system can never provide expected QoE.

A. Formal Definition of Perceived Reliability

Define the global video (or audio) system state at time t as a set of stream states at t, one for each video (or audio) stream. A global state is valid if it can actually occur during a performance of duration d. We partition the entire space of valid global audio states into n audio levels A_i, such that the levels represent progressive quality degradation. For example, A_3 represents greater audio degradation compared to A_2. Likewise, we produce a partition of m progressively degraded video levels V_j, resulting in a Cartesian product of $n \times m$ total quality levels. We refer to those levels as $Q_{ij} = (A_i, V_j)$.

For each A_i and V_j we define a boolean predicate B_{A_i} or B_{V_j} operating on global audio or video states that returns true iff the state belongs to A_i (or respectively V_j). Additionally, for each level $A_i, i > 1$ and $V_j, j > 1$, we define tolerable duration θ_{A_i} or θ_{V_j}. The durations are sorted in the decreasing order. For example, $\theta_{A_2} > \theta_{A_3}$. Finally, we denote perceptibility threshold μ_A and μ_V for audio and video, respectively. Section VIII briefly discusses how we choose meaningful values for μ_A and μ_V in the context of World Opera.

Given an audio level A_i, we now define the criterion for A_i that the performance needs to meet.

1) *Instant level* at time t is A_i if B_{A_i} evaluates as true when applied on the global audio state at time t. Since the predicates are mutually exclusive, and collectively they cover the entire space of system states, there is one and only one predicate that evaluates as true at any given moment.

2) *Perceived level* at time t is defined as the worst of all instant levels during the entire recent interval $[t-\mu_A, t]$, except when $t < \mu_A$, for which it is equal to the instant level. The case of $t < \mu_A$ does not play an important role in practice because the typical performance duration $d \gg \mu_A$.

TABLE I: Stages considered in the analyzed scenario, and WO elements located in each of them.

Stage A	Orchestra	Singer	Harpsichord	*Audience*
Stage B	Orchestra	Singer	*Audience*	
Stage C	Orchestra	Conductor	*Audience*	

3) For each perceived level A_i, we consider the total time a given performance delivers A_i, i.e., the cumulative time that the perceived level is A_i. Formally, $T_{A_i} = \int_{t=0}^{d} f(t)\,\mathrm{d}t$, where $f(t)$ is equal to 1 if the perceived level at t is A_i, and to 0 otherwise.

4) The performance exceeds tolerable duration θ_{A_i} for perceived level $A_i, i > 1$ if the total time the performance delivers A_i, or a worse level, exceeds θ_{A_i}, i.e. if $\sum_{j \geq i} T_{A_j} > \theta_{A_i}$.

5) The performance satisfies the *tolerability criterion* for perceived level A_i if for each level $A_j, j > i$, the performance does not exceed tolerable duration θ_{A_j}. Note that if the performance satisfies the tolerability criterion for A_i, it will satisfy the tolerability criterion for all levels $A_j, i < j \leq n$. The performance always satisfies the tolerability criterion for A_n.

The criterion for V_j is defined similarly. *Perceived reliability* for QoE level Q_{ij} is the probability that the performance meets the criteria for both A_i and V_j. It should be noted that classic reliability is a lower bound for perceived reliability: certain configurations that are considered incorrect service in the former case, define instead correct (perceived) service in the latter. More precisely, classic reliability can be seen as perceived reliability where all the perceptibility thresholds μ_X and tolerable durations θ_X are zero.

The above definition is given from the standpoint of a single given user. In order to account for the diversity of user requirements explained in Section IV-B, different sets of QoE levels (and even different perceptibility thresholds) may need to be defined for different users. For a realistic large-scale TI application, this may result in a large number of different QoE levels. In order to make the task of defining a dependability model more manageable and the evaluation more scalable, we propose to define groups of users and to group QoE requirements together (see Figure 3). For example, all the singers on the same stage may have the same expectations from the performance. The audience can also be divided into groups, based on individual location or human characteristics. This division is much more fine-grained compared to the separation between the artists and audience typical in existing QoE evaluations, as discussed in Section IV-B.

VI. Integrating QoE in the World Opera Scenario

This section introduces a concretization for the QoE levels introduced in Section V. We start with describing a WO scenario and then present the QoE and dependability requirements for individual user roles involved in the scenario. The level definitions are derived from the artistic content and user roles; individual human characteristics such as mood are left for future work.

A. Scenario Description

The WO experimental performance [19] is intended for three stages, A, B and C, hosting different classes of artists as

presented in Table I. The conductor needs to hear the music from the harpsichord and all orchestras, in addition to the singers' voices. If one singer is synchronized with the conductor, another singer can synchronize through the first singer. Hence, the conductor can afford to lose the voice of one of the singers for a short period of time. Furthermore, it is important for the conductor to watch the orchestra and at least one of the singers in order to coordinate the ensemble.

Audiences located at all stages need to see local and remote artists together. They want to hear the synchronized rhythm of singers', orchestras, and the harpsichord music. Depending on the type of performance, the audience may tolerate a brief loss of audio and video from some of the artists.

Singers wants to hear the music from all stages and must see the conductor (either locally or through the video) to keep synchronized among themselves as well as with the orchestra and harpsichord. However, it is possible for a singer to perform even if the video and audio from another singer or from the harpsichord are unavailable for a short time period [19].

The orchestra at each stage needs to hear their remote counterparts. Further, the orchestra itself must remain in sync even in the case of a brief loss of audio from singers' and the harpsichord. It is also important for the orchestra to watch visual cues given by the conductor in order to remain in sync with the other artists. Additionally, it is useful for the orchestra to watch the facial expressions of singers. Depending on the type of performance, the orchestra may tolerate the loss of video streams from some of the artists (other than the conductor) for a limited duration.

In order to remain in sync with other artists, the harpsichord player needs to hear the singers' voices or remote music from the orchestra, in addition to watching the visual cues given by the conductor. Furthermore, it is useful to watch the singers and orchestra for better experience.

B. QoE and Dependability Requirements for the Scenario

Table II describes QoE levels for the roles of conductor, audience, singer, orchestra, and harpsichord. For each role, we consider three QoE levels for the audio subsystem, A_1, A_2 and A_3, and three QoE levels for the video subsystem, V_1, V_2 and V_3. Combining these audio and video levels yields 9 possible QoE levels for the WO performance. $\hat{R}_{ij}$ refers to perceived reliability for audio quality level i, and video quality level j.

The scenario and requirements presented in this section and specifically, in Table II are used for our evaluation case study in Section VIII. While World Opera is an actual application driven by the artistic consortium today, evolution of TI is still in its early stages. We believe that the same approach is applicable to modeling of large-scale and even more sophisticated TI applications that will emerge in the future.

TABLE II: Quality levels for conductor (CN), audience (AU), singer (SI), orchestra (OA), and harpsichord (HA). $\#(k)$ is the number of correct streams of type k, while $T(k)$ is the total number of streams of type k. We define two helper predicates: perfect(k) := ($\#(k) = T(k)$) and present(k) := ($1 \leq \#(k)$). θ_{Level} is the tolerable duration for the given quality level.

Level	Predicate B_{Level} for conductor	θ_{Level}
A_1	$\forall k \in \{OA, SI, HA\}$: perfect(k)	—
A_2	$\forall k \in \{OA, HA\}$: perfect(k) $\wedge$ present(SI) $\wedge \neg B_{A_1}$	2 sec
A_3	$\neg B_{A_1} \wedge \neg B_{A_2}$	0.5 sec
V_1	$\forall k \in \{OA, SI, HA\}$: perfect(k)	—
V_2	perfect(OA) $\wedge$ present(SI) $\wedge \neg B_{V_1}$	5 sec
V_3	$\neg B_{V_1} \wedge \neg B_{V_2}$	1 sec
	Predicate B_{Level} for audience	θ_{Level}
A_1	$\forall k \in \{OA, SI, HA, AU\}$: perfect(k)	—
A_2	$1 \leq \lvert\{k \in \{OA, SI, HA\} : \text{perfect}(k)\}\rvert \leq 3$	2 sec
A_3	$\neg B_{A_1} \wedge \neg B_{A_2}$	0.5 sec
V_1	$\forall k \in \{OA, SI, HA, AU, CN\}$: perfect(k)	—
V_2	$2 \leq \lvert\{k \in \{OA, SI, HA, CN\} : \text{perfect}(k)\}\rvert \leq 4$	5 sec
V_3	$\neg B_{V_1} \wedge \neg B_{V_2}$	1 sec
	Predicate B_{Level} for singer	θ_{Level}
A_1	$\forall k \in \{OA, SI, HA\}$: perfect(k)	—
A_2	$\lvert\{k \in \{OA, SI, HA\} : \text{perfect}(k)\}\rvert = 2$	2 sec
A_3	$\neg B_{A_1} \wedge \neg B_{A_2}$	0.5 sec
V_1	$\forall k \in \{OA, SI, HA, CN\}$: perfect(k)	—
V_2	$\lvert\{k \in \{OA, SI, HA, CN\} : \text{perfect}(k)\}\rvert = 3$	5 sec
V_3	$\neg B_{V_1} \wedge \neg B_{V_2}$	1 sec
	Predicate B_{Level} for orchestra	θ_{Level}
A_1	$\forall k \in \{OA, SI, HA\}$: perfect(k)	—
A_2	perfect(OA) $\wedge$ $\lvert\{k \in \{SI, HA\} : \text{perfect}(k)\}\rvert = 1$	2 sec
A_3	$\neg B_{A_1} \wedge \neg B_{A_2}$	0.5 sec
V_1	$\forall k \in \{OA, SI, HA, CN\}$: perfect(k)	—
V_2	perfect(CN) $\wedge$ $\lvert\{k \in \{OA, SI, HA\} : \text{perfect}(k)\}\rvert = 2$	5 sec
V_3	$\neg B_{V_1} \wedge \neg B_{V_2}$	1 sec
	Predicate B_{Level} for harpsichord	θ_{Level}
A_1	$\forall k \in \{OA, SI\}$: perfect(k)	—
A_2	$\lvert\{k \in \{OA, SI\} : \text{perfect}(k)\}\rvert = 1$	2 sec
A_3	$\neg B_{A_1} \wedge \neg B_{A_2}$	0.5 sec
V_1	$\forall k \in \{OA, SI, CN, AU\}$: perfect(k)	—
V_2	perfect(CN) $\wedge$ $\lvert\{k \in \{OA, SI\} : \text{perfect}(k)\}\rvert = 1$	5 sec
V_3	$\neg B_{V_1} \wedge \neg B_{V_2}$	1 sec

VII. Evaluation Framework

We now define a framework for reliability and QoE evaluation of distributed TI applications, and apply it to the WO use-case. In our previous work [20], we evaluated the perceived QoE of a single WO stage. In this paper, we extend the framework to support the modeling of multiple stages, including failure propagation between them, support for different stage configurations, and for the modeling of communication links.

Given a QoE level Q_{ij}, the framework supports the evaluation of the following metrics, for different classes of users:

$R_{Q_{ij}}(t)$**:** *reliability* for quality level Q_{ij}, i.e., the probability of delivering quality level Q_{ij} or above, up to time t;

$T_{Q_{ij}}(t)$**:** *amount of time* delivering level Q_{ij} in $[0, t]$;

$\hat{R}_{Q_{ij}}(t)$**:** *perceived reliability* for level Q_{ij} (see Section V).

Classic reliability is included here for comparison with perceived reliability.

A. Modeling Approach

One of the main challenges in modeling TI applications is the large number of components and their complex interactions. We overcome this problem by using modularity: the system model is realized by composing a set of elementary submodels, addressing different aspects of the system. Particular attention

is devoted to the identification of the *interfaces* and *parameters* of the different submodels, allowing them to be modified in isolation from the rest of the model, thus facilitating the evaluation of different system configurations, and improving the extensibility of the framework (e.g., introducing new component types like haptic sensors).

To support this process, we rely on "template" submodels (see [21], [22]), which are then instantiated multiple times, with different parameters settings, and connected through their interfaces. This approach saves the model designer from having to manually create (and maintain) multiple models for components with similar behavior. Also, any change in a template model is automatically reflected in all template instances, thus allowing us to manage very large TI systems.

We have realized our approach using Stochastic Activity Networks (SANs) [5], a formalism that extends Stochastic Petri Nets (SPNs) [23]. The Möbius framework [6] provides useful primitives for implementing our approach. SAN models are composed using the Replicate/Join state-sharing formalism [24]: the *Join* composition operator is used to compose two or more submodels, by sharing places between them; the *Replicate* composition operator is used to combine multiple identical copies of a submodel, also sharing places between them. To define model interfaces and parameters, special kinds of places can be added to SAN models, called "extended places", which can contain complex data types.

B. Model Templates for the World Opera Application

To model the WO application we define the following set of basic model templates:

- `Component`, which models a single functional component of the WO architecture;
- `StreamAcquiring`, which models the process of capturing a stream from a set of acquiring components;
- `StreamProcessing`, which models the processing of a single stream by a processing component;
- `StreamMixing`, which models the mixing of a number of input streams into a number of output streams.

We describe them in the following, including their interfaces and parameters, taking the WO system as reference. Further details are available in a technical report [25]. It should be noted that, depending on the failure assumptions being considered, the actual implementation of template models can change; as such, the following sections describe one of the possible implementations of such templates. On the other hand the approach is general: the same set of templates, with the same interfaces and composition rules, can be adopted for QoE evaluation of different TI applications.

1) Component: The `Component` template model represents a single functional component of the WO architecture, possibly using some failure handling mechanism. The model has three interface places: `Failed`, `CountGroup` and `CountFailed`. The `Failed` place represents the current working state of the functional component: it contains a token if the component has failed, otherwise it is empty. Places `CountGroup` and `CountFailed` are used to track a set of components belonging to the same logical group, e.g., an array of microphones

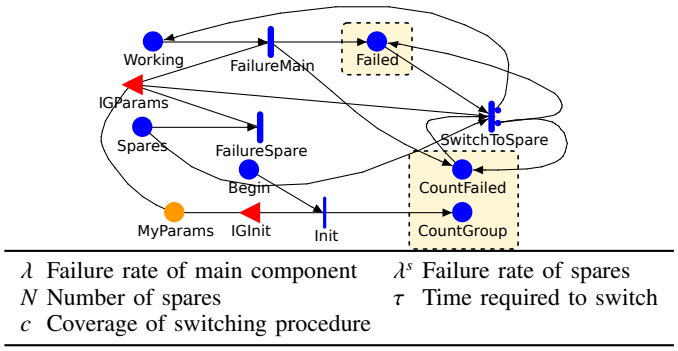


Fig. 4: `Component` template model and its parameters.

capturing the same audio stream. They hold the total number of components in the group, and the number of them that are currently failed, respectively.

Having specified those interfaces, the internal behavior of the model depends on the failure assumptions, and on the adopted failure handling mechanisms. Figure 4 depicts a template model for a functional component using a hot-spare failure handling mechanism; the total number of available spares is given by the N parameter of the template. Other parameters of the template include: i) the probability c to successfully switch to a spare component, leaving the probability $(1 - c)$ to abort the switch because of losing the spare; ii) the delay τ required to perform the switch, assumed to follow an exponential distribution; iii) the failure rate λ of the active component; and iv) the failure rate λ_s of spare components. Note that both automated and manual switching mechanisms can be modeled, by increasing or decreasing the τ parameter.

2) StreamAcquiring: The `StreamAcquiring` template models the capturing of one multimedia stream from a set of acquiring components. It has three interface places.

Place `CountGroup` contains the number of components from which the multimedia stream is being captured, e.g., an array of microphones; place `CountFailed` count the number of such components that are currently in a failed state; finally, place `StreamStateOut` contains the state of the captured stream. In general, this template model realizes a mapping between the state of acquiring components and the state of the stream, i.e., it updates the marking of place `StreamStateOut` based on the marking of the other two interface places.

Figure 5 shows the `StreamAcquiring` template model for the scenario modeled in this paper, where streams can be in two different states: *nominal* and *missing*. A stream is in nominal state when the number of correctly working acquiring components is above a given threshold; it is considered missing otherwise. Such threshold can be specified as a numerical value (γ^k parameter), or as a percentage of the total number of components that are capturing the stream ($\gamma^{\%}$ parameter).

The model contains an activity for each possible state of the stream, which is used to update the marking of `StreamStateOut` when the state of acquiring components changes. For example, the `StreamMissing` activity in Figure 5 fires when i) the number of tokens in `CountFailed` is above the threshold, and ii) place `StreamStateOut` contains a token, representing the nominal state. The firing of this transition removes the token from `StreamStateOut`, thus

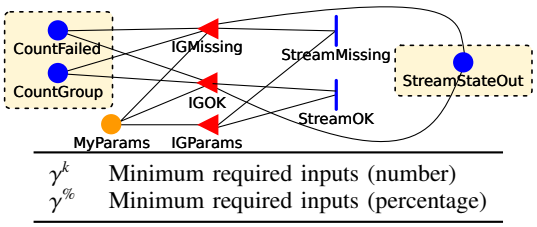


Fig. 5: `StreamAcquiring` template model and its parameters.

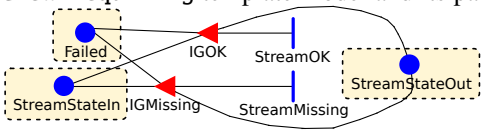


Fig. 6: `StreamProcessing` template model.

propagating the failure of components as a stream failure.

3) StreamProcessing: The `StreamProcessing` template models a component's processing of a stream. By the term "processing" we mean any action that involves receiving a multimedia stream as input and producing an output stream. Thus, also the gateway is considered a processing component. This template model has three interface places. Place `Failed` contains the current state of the component that is in charge of processing the stream; a token indicates that the component has failed. Places `StreamStateIn` and `StreamStateOut` contain the state of the stream as received by the component, and as produced as output, respectively. The model performs a mapping from: the state of the received stream and the state of the processing component, to the state of the stream produced as output. Figure 6 depicts the implementation of the model for the target scenario.

4) StreamMixing: Sometimes, a set of multimedia streams are mixed together, to produce a single output stream. This behavior is modeled by the `StreamMixing` template model [25], which is model is very similar to the previous one, and it is not described here for the sake of brevity.

C. Template Composition — Single Stage

The model of a WO stage is obtained by properly selecting instances of the previously introduced template models, and connecting them according to the stage architecture. While the resulting model is different for every stage configuration, there are some recurring "patterns", which are summarized in the following. The application of these patterns can be automated.

1) For each functional component in the stage, a `Component` template instance is added.
2) For each stream that is *acquired* in the stage, an instance of the `StreamAcquiring` template is added.
3) For each (p, s) such that component p processes stream s, an instance of the `StreamProcessing` template is added.
4) To model a set of streams $\{s_1, \ldots, s_n\}$ which is mixed to a single output stream s', an instance of the `StreamMixing` template is used.

The obtained template instances are then combined by state-sharing, connecting their interface places based on the path followed by multimedia streams within the architecture.

Figure 7 shows an example of a simplified WO stage architecture. The example considers a show consisting of two audio (A_1, A_2) and two video (V_1, V_2) streams. Two of them (A_1 and V_1) are acquired locally, while the other two are received from another stage. The path of streams across the stage is depicted in Figure 7a. Let us focus on video stream V_1; the stream is captured by one or more cameras and processed by a camera workstation. It is then forwarded i) to the gateway, to be transmitted to the other stages, and ii) to a display workstation to be reproduced locally. After being processed by the display workstation, the stream is finally routed to one or more projectors to be displayed to the audience.

Assuming that a `Component` instance has been added for each functional component of the WO stage, the path of stream V_1, is represented in the model as follows. First, a new `StreamAcquiring` instance, `sc_`V_1, is added to the model; its interfaces `CountGroup` and `CountFailed` are then connected to the corresponding interfaces of each camera `Component` responsible for capturing stream V_1. Then, three `StreamProcessing` instances are added: `sp_wsc_`V_1 for the camera workstation, `sp_wsd_`V_1 for the display workstation, and `sp_gw_`V_1 for the gateway.

The `StreamStateOut` interface of `sc_`V_1 is connected to the `StreamStateIn` interface of `sp_wsc_`V_1, i.e., the stream state produced by cameras is the same as the one received as input by the camera workstation. The same procedure is repeated for the rest of the path of stream V_1 in the stage architecture. Finally, the `StreamStateOut` interface of `sp_gw_`V_1, i.e., the state of stream V_1 at the output of the gateway, becomes an interface of the overall stage model.

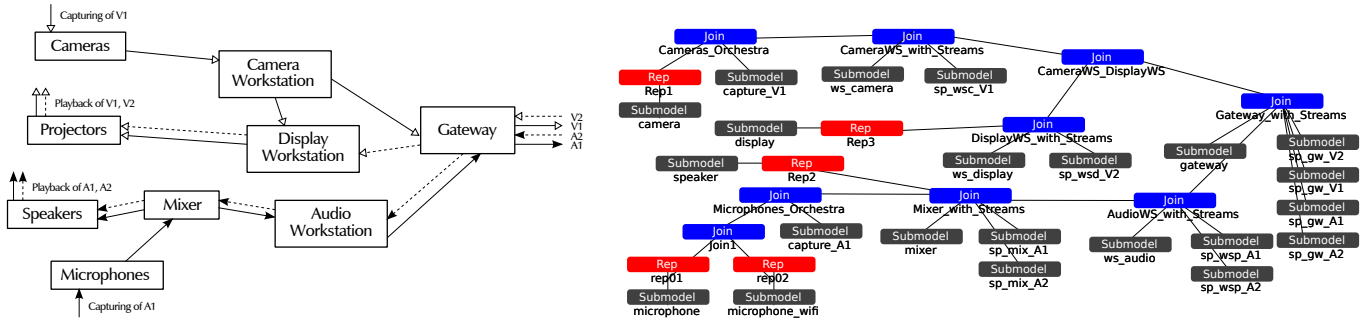
The resulting composed model for the discussed example is depicted in Figure 7b. It should be noted that the stage model obtained by composing the basic template models has strong structural relations with the physical stage architecture, and with the flow of streams across stage components.

This composition approach, which is one of the key features of our framework, further facilitates the automated construction of the analysis model, based on a schematic description of stage architectures and model transformation (e.g., see [22]).

D. Template Composition — Multiple Stages

Once the models for individual WO stages have been obtained, they are connected to obtain the global model of a WO performance. If the models of individual stages have been built using the procedure described in the previous section, they should include an interface place for each of the multimedia streams that compose the WO application. In particular, the model of each stage should have a $\texttt{StreamStateOut}_\text{i}$ place for each of the streams that are captured in that stage, and a $\texttt{StreamStateIn}_\text{j}$ place for each multimedia stream that is received from a remote stage.

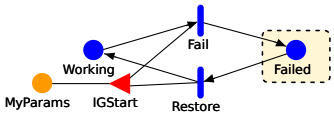
In our previous example, the stage would have four interfaces: A_1`_out` and V_1`_out`, corresponding to the `StreamStateOut` places of `StreamProcessing` models of the gateway for streams V_1 and A_1; and A_2`_in` and V_2`_in`, corresponding to the `StreamStateIn` places of `StreamProcessing` models of the gateway for streams that are received from other stages (V_2 and A_2). The models for the different stages are then connected by matching each



(a) Simplified example of a WO stage architecture.

(b) The corresponding composed SAN model.

Fig. 7: A simplified WO stage model for a show consisting of two video (V_1, V_2) and two audio (A_1, A_2) streams. The composed SAN model for an individual stage (b) can be constructed automatically from a logical diagram of the stage architecture (a).



λ	Link failure rate
μ	Link repair rate

Fig. 8: `NetworkLink` template model and its parameters.

`StreamStateOut` place of the stage where the stream is captured, with the `StreamStateIn` places corresponding to the same stream in the other stages (see [25] for details).

E. Modeling Communication Links

Communication links are simply modeled as *processing components*. To this extent, we add the `NetworkLink` template, as a special variant of the *Component* template modeling communication links. A simple on/off implementation of this model is depicted in Figure 8. The model contains the single `Failed` interface, and it is characterized by two parameters: λ, the failure rate of the network, and μ, its recovery rate.

Thus, to model an unreliable link between two processing components, it is sufficient to add an instance of the `NetworkLink` model, and an additional `StreamProcessing` instance for each stream transmitted through the link. This demonstrates the modularity and flexibility of the framework, which can be extended to model additional components (e.g., routers) in a uniform way.

F. Summary and Extensions

In the previous sections we described the core of our framework for modeling TI applications. With this framework, it is easy to modify and extend our models.

Different failure handling mechanisms (instead of hot-spares) can be considered, with different implementations of the `Component` template model. While our solution considers only fail-stop behavior, other failures modes could also be considered for components, application streams, or both. This can be done by encoding different failure modes in terms of the number of tokens in the `Failed` place, and modify the models accordingly. For example, one token could model a "stop" failure, and two tokens could model a "noisy" failure.

Delays and other more complex failure propagation behaviors can be modeled, with different implementations of the `StreamProcessing` template. For example, the output stream from a display workstation may fail, when no data is received as input for a given period of time.

Such modifications can be progressively applied to refine the metrics and/or to analyze specific aspects of a WO show. Here we focus on the effect of failure propagation on QoE metrics, and leave these extensions for future work.

Finally, we note that the proposed framework is applicable to WO configurations of arbitrary size, in terms of number of stages and number of components per stage. The model for each stage is constructed using the procedure described in Section VII-C, regardless of the number of components involved; any number of stage models can then be composed together following the procedure in Section VII-D. Metrics evaluation relies on additional deterministic transitions to model the elapsing of tolerable durations, and rewards structures defined based on Table II. Further details can be found in [25].

VIII. Practical Application and Results

Next, we build and evaluate a model for the WO scenario, based on the framework described in Section VII. The main goal of our evaluation is to understand the architectural choices that maximize the QoE of users in presence of failures in TI applications. As part of this main goal, we have identified the following subgoals: i) understanding the impact of component failures and the recovery mechanisms on the QoE perceived by various users; and ii) comparing the proposed perceived reliability metric with the traditional reliability metric.

We used the discrete-event simulator provided with the Möbius framework [6] for the evaluation. Computing an analytical solution is not feasible, since our model allows multiple deterministic activities to be enabled at a time [23]; this is necessary to correctly account for the "unperceivable failures" and "tolerable duration" intervals for the different quality levels. Even if we had considered exponential distributions only, the state-space explosion problem would have prevented the application of analytical solution techniques. All the values have been computed from at least 10^3 simulation batches, with a confidence level of at least 99%, and a confidence half-interval of 10^{-3} or smaller. The actual maximum error in the computed results was $\pm 8.24 \cdot 10^{-4}$, i.e. more than three orders of magnitude smaller than the results. We present the errors in Figure 10 and Figure 13 because the differences between the results in these plots are very small, so that the errors affect the relative comparison and support the analysis of results. In all the other plots, the errors do not affect the conclusions, they would not be visible, and are thus omitted.

TABLE III: Main model parameters and their default values.

Component type	λ ($hours^{-1}$)	c	τ (secs)	$\gamma^{\%}$
Camera	$2 \cdot 10^{-3}$	0.95	60	50%
Workstation – Camera	$1 \cdot 10^{-5}$	0.95	180	-
Workstation – Display	$1 \cdot 10^{-5}$	0.95	180	-
Workstation – Audio	$1 \cdot 10^{-5}$	0.95	5	-
Projector	$6 \cdot 10^{-3}$	0.95	60	50%
Speaker	$1 \cdot 10^{-3}$	0.95	1	50%
Microphone – Wired	$2 \cdot 10^{-3}$	0.95	5	50%
Microphone – Wireless	$2 \cdot 10^{-3}$	0.95	120	50%
Mixer	$1 \cdot 10^{-4}$	0.95	5	-
Gateway	$1.19 \cdot 10^{-6}$	0.95	5	-
Network	$\lambda = 1 \cdot 10^{-5}$ hours^{-1}		$\mu = 1/180$ secs^{-1}	
Unperceivable Durations	*Audio:* 10 ms		*Video:* 80 ms	

A. The Analyzed Scenario

For our evaluation, we consider the scenario discussed in Section VI and use the corresponding quality levels defined in Table II. This scenario is based on the past World Opera experiments and interviews conducted during those experiments, which established the relative importance of different video and audio cues (and correspondingly, streams) [19]. Unfortunately, those experiments did not consider the tolerable failure duration parameter θ. In order to compensate for that, we perform sensitivity analysis with respect to the θ parameter; Table II presents the default values for θ.

Perceptibility threshold for audio and video streams in the context of WO is a subject of active research [26], [27], [19]. Following the findings of these works, our evaluation uses the perceptibility threshold values of 10 ms and 80 ms for audio and video streams, respectively, for all user roles and stages.

Table III summarizes the main model parameters and their default values used in our evaluation. The failure rates are conservative estimates [28], [29]. Note that microphone failure rates include poor connections and operation issues; the same applies to other hardware devices. Besides, during certain WO productions, the artists wear custom camouflaged microphones, which are significantly less reliable in operation.

The switching time and coverage (τ and c, see Section VII), and the threshold $\gamma^{\%}$ are our conservative estimates based on discussions with WO technicians. While our model is independent of the failure distribution, we assume that failure rates are exponentially distributed, following conventional practice. For all experiments, we fix the performance duration d to 2 hours, representative of most Opera performances. As these systems are typically deployed over dedicated network infrastructures, we assume network links have a low failure rate.

B. Results

1) Reference Scenario: We first analyzed the reference scenario, as defined by the parameters in Table III. Figure 9 shows the perceived reliability, $\hat{R}(t)$, experienced by different classes of users located at the three WO stages, comparing results where different minimal acceptable quality levels are assumed: A_1V_3 ("perfect" audio, with no requirements on the video); A_3V_1 ("perfect" video, with no requirements on audio), and A_1V_1 ("perfect" quality for both audio and video).

Based on the results shown in Figure 9, we can draw a few conclusions. First, the perceived reliability is different for the three stages. This aspect is evident when comparing the results for stage A with those for stages B and C; however, minor differences also exist between stages B and C. Considering $\hat{R}_{A_1V_1}$ for the audience, the result in stage A is 0.0339 higher than in B, which in turn is 0.0067 higher than in C. Note that those differences are much higher than the confidence interval, and are thus not caused by low accuracy of simulations.

These differences are caused by the different arrangements of artists with respect to the overall performance; in fact, it should be noted that similar components in different stages are assumed to have the same physical properties (e.g., failure rate). The audience at stage A has the advantage of being co-located with two key elements of the WO performance, namely one of the two singers and the harpsichord. Similarly, since the other singer is located at stage B, its audience experiences a slightly higher quality compared to the audience at stage C.

We also note that the audience experiences a lower perceived reliability than artists, with this result being consistent across all three stages. The difference is higher for video than for audio. The reason for this behavior is in the quality levels considered for the audience: the highest quality level, A_1V_1, is delivered only when most of the application streams are in the correct state, including audio and video streams of remote audiences, which are not necessary for artists. The audience is therefore more exposed to failures during the WO performance

2) Adding Spares: Next, we study the impact of adding spares to components of the WO architecture. The results are shown in Figure 10. We focus on the impact of adding spares for every component of the architecture versus adding spares only for the least reliable components, i.e., microphones, speakers, cameras, and projectors. Our results extend those obtained in [20] for the single-stage scenario: adding spares for all components, including workstations and gateways, does not provide any improvement with respect to adding spares for the least reliable components only. Furthermore, adding two spares does not provide much advantage either. We observe that for all the experiments with spares, the confidence intervals of results are overlapping, leading to the conclusion that the small differences that are visible in the figure are due to simulation errors.

3) Microphone Failure Rates: The data reported in Figure 11 analyzes the effect of varying the failure rate of microphones, comparing the impact of wired and wireless microphones. A similar evaluation was performed in [20], for a single-stage scenario, suggesting that varying the failure rate of wired microphones had a negligible impact on perceived reliability, mainly due to the short time required to switch to a spare microphone (see Table III). We further inspect this behavior in Figure 11, by comparing the impact of failure rates of wired and wireless microphones on the perceived reliability experienced by different users of the system. While the results partially confirm the trend observed in [20], we also notice that a higher failure rate for wired microphones reduces the perceived reliability for the audience of stages B and C, while it does not affect the audience of stage A or the artists. The impact of this architectural variant is considerable, as it

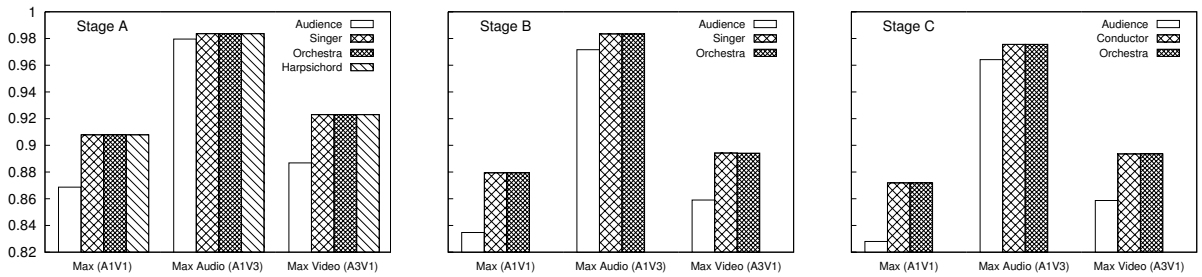


Fig. 9: Perceived reliability for different quality levels and users, at each of the three stages.

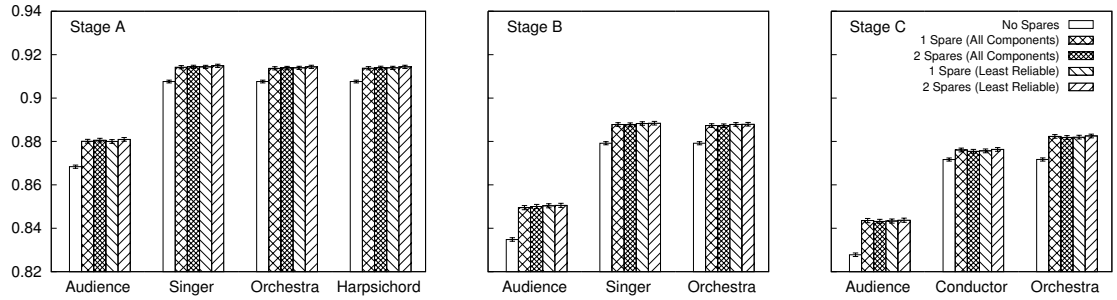


Fig. 10: Perceived reliability for the highest quality level, $\hat{R}_{A_1V_1}$, when varying the number of spares for each component.

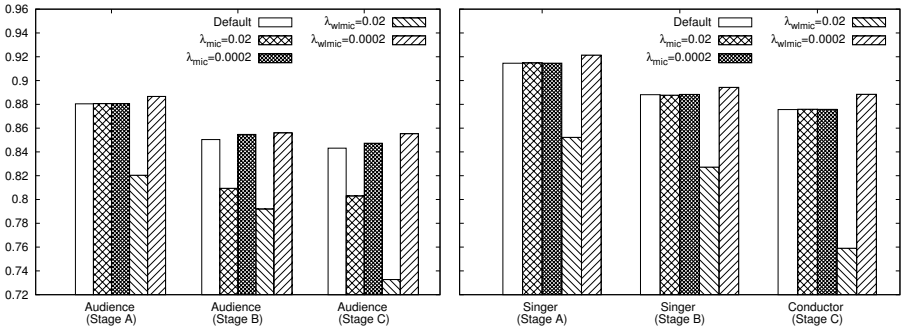


Fig. 11: Perceived reliability for the highest quality level, $\hat{R}_{A_1V_1}$, when varying the failure rate of microphones.

consists in a reduction of 0.04 in perceived reliability, i.e., 4%.

To understand this behavior, recall that, as opposite to artists, the quality level for the audience is also affected by the audio streams of remote audiences, captured using wired microphones. These additional microphone-dependent streams increase the probability of reaching the A_2 level due to a microphone failure. Furthermore, while running with the A_2 level, additional microphone failures may occur, eventually reaching the A_3 level. Even though switching to spares is a fast operation for wired microphones, the tolerable duration for A_3 is quite short, which increases the chance of not respecting the tolerability criterion. The low impact of wired microphones on the audience at stage A is due to the harpsichord being also located at stage A, thus reducing the number of application streams affected by failures of wired microphones.

4) Switching Time: In Figure 12 we explore how the switching time affects the perceived reliability of users at stages A and C. We compare the results obtained using nominal values for τ (Table III) with those obtained by reducing τ by a factor of 10 and 100. Such improvements can be obtained by devising automated switching mechanisms.

Overall, reducing the switching time improves the perceived reliability. However, the impact is much higher for video (average improvement of 0.039 when reducing τ to $\tau/10$) than for audio (average improvement of 0.006). This is because audio components already have low switching time. Another observation is that the perceived reliability for the conductor (who is at stage C) is not affected by the switching time of audio components: the observed improvement from τ to $\tau/10$ is only 0.0000055, much smaller than the confidence interval for the obtained results, and thus not relevant. This is due to the definition of quality levels for the conductor: even A_2 requires the harpsichord stream to be in the correct state, which means that the unavailability of the harpsichord stream immediately brings the perceived audio level to A_3. The same is not observed for the perceived reliability of video, since the harpsichord video stream is excluded from the conductor's defined quality levels. Finally, it should be noted that improving the switching time has a greater impact on the audience than on artists. Carefully examining such differences and the perceived reliability requirements of users and artists, could help stage technicians decide which components need a switching mechanism.

5) Impact of the Tolerable Duration: The results shown in Figure 13 analyze the impact of the "tolerable duration" parameter (θ) associated with the different quality levels. In our QoE framework, such parameters can be used to model the sensitivity of users to the delivery of degraded quality

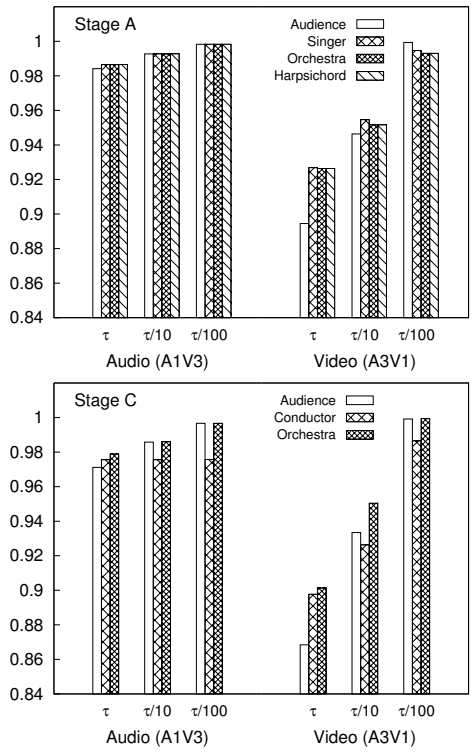


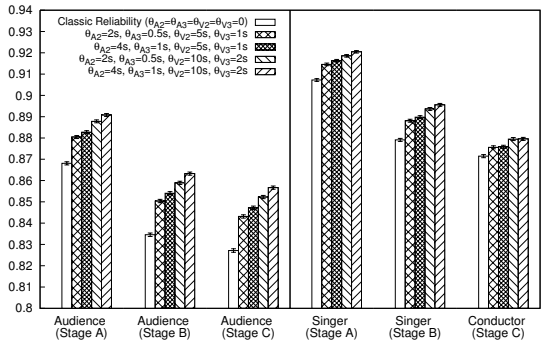
Fig. 12: Effect of the switching time on perceived reliability

levels. Variations of such parameters can be applied in order to, e.g., reflect the subjectivity of different persons, or the characteristics of a particular show. The evaluation considers a scenario where all the components are redundant with two spares. For reference, the results are also compared with "classical" *reliability*. In these plots, error bars are explicitly displayed, to highlight variations that are statistically relevant and identify those that can be regarded as simulation errors.

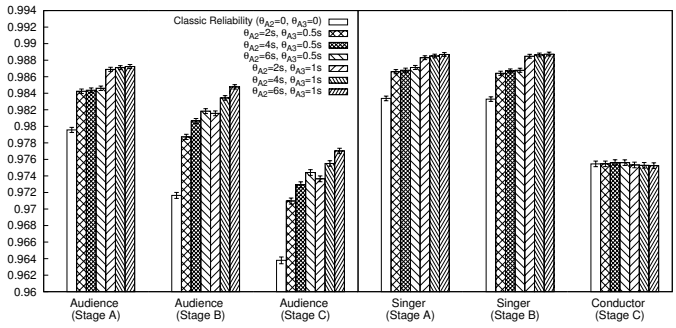
Figure 13a depicts the perceived reliability for the highest quality level, $\hat{R}_{A_1V_1}$, while varying the values of θ_{A_2}, θ_{A_3}, θ_{V_2}, and θ_{V_3}. Specifically, the default values for these parameters presented in Table II were doubled in some of the experiments. As expected, the perceived reliability is always higher than the classic reliability. The difference becomes even more pronounced for bigger values of θ. While this is observable for all users in the system, the improvement is higher for the audience than the artists. This is because the audience relies on all the streams of the WO performance, and thus gains much more from allowing longer periods of service interruption.

The two subsequent plots, Figure 13b and Figure 13c, further study the impact of the tolerable duration parameters, by analyzing the audio and video subsystem individually. Figure 13b shows results for A_1V_3 (i.e., the user wants perfect audio quality, but has no requirements on video quality), while Figure 13c shows results with respect to level A_3V_1 (i.e., the user wants perfect video quality, but has no requirements on audio quality). In these plots, an even wider variation in tolerable duration values is used.

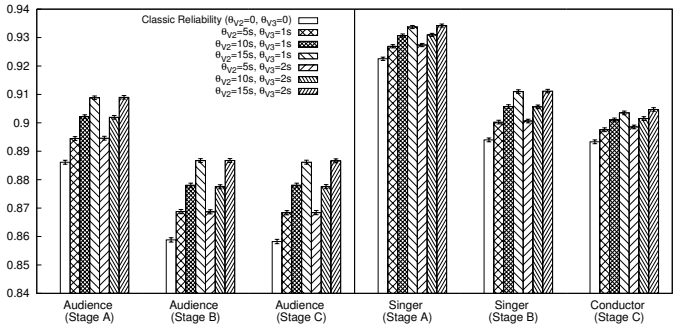
We observe the following trends: First of all, the more significant threat to the perceived reliability comes from the video subsystem. In the considered scenario, the perceived reliability considering only audio quality, $\hat{R}_{A_1V_3}$, is always



(a) Highest quality level for different users ($\hat{R}_{A_1V_1}$)



(b) Highest audio level for different users ($\hat{R}_{A_1V_3}$)



(c) Highest video level for different users ($\hat{R}_{A_3V_1}$)

Fig. 13: Perceived reliability for different tolerable durations

higher than 97% (Figure 13b), while the corresponding one for video, $\hat{R}_{A_3V_1}$, remains below 94% even after increasing θ_{V_2} and θ_{V_3} (Figure 13c). Second, perceived reliability of the video subsystem is highly affected by the tolerable duration for level V_2 (θ_{V_2}). This is because most video components have long switching times: When level V_2 is reached, the probability to recover within the acceptable time is very low; therefore, increasing θ_{V_2} provides a significant improvement. Conversely, the tolerable duration θ_{V_3} for level V_3 is practically irrelevant. This means that the probability of reaching such level before exceeding θ_{V_2} is low (i.e., the tolerable duration is more likely to be exceeded for V_2 rather than V_3).

The audio subsystem has the opposite behavior: perceived reliability increases only when the tolerable duration θ_{A_3} with respect to A_3 is increased. On the other hand, the tolerable duration θ_{A_2} with respect to A_2 is mostly irrelevant. This means that, most of the time, recovery from A_2 is possible within the acceptable timings, since most audio components have a low delay for switching to the spare (see Table III). However, multiple failures may lead directly to A_3, from which timely recovery is very difficult; increasing the tolerable duration

parameter for θ_{A_3} has thus impact.

Exceptions to the above behavior can be observed for the conductor at stage C and for the audience at stages B and C. For the conductor, the value of tolerable duration for A_2 is irrelevant (Figure 13b). As previously noted, this is due to the definition of quality levels for the conductor: the perceived level directly becomes A_3 upon the failure of the harpsichord stream. As opposed to the audience at stage A, the amount of time during which the audience at stages B and C can tolerate level A_2 has a significant impact on the metrics. This result is due to the arrangement of artists across the stages: recall that two key components of the WO performance, the harpsichord and one of the two singers are located at stage A, and thus their audio is transmitted remotely to stages B and C.

C. Summary of Results

The following summary of our results holds for the multiple stage scenario of WO analyzed in this paper. Figure 9 shows that the perceived reliability values are different at different locations, and for different user roles, despite all stages having the same architecture and components. It also proves that appropriate paring of artists at one location (for example, a singer and a harpsichord) increases the perceived reliability for the audience at that location. This information helps prioritizing the stages/locations in terms of perceived reliability.

Figure 10 confirms that adding spares for all components does not provide a significant improvement as compared to adding spares for the least reliable components only. Figure 11 indicates that, depending on the performance characteristics, using highly reliable components does not necessarily improve the performance for all user roles at all stages. Results in Figure 12 allow the engineers to compare different procedures for switching to spares (e.g, automated or manual). In general. a faster failover mechanism improves perceived reliability, but it also depends on the profile of the user (e.g., the conductor). Such results allow engineers to compare different stage architectures to tune the dependability budget.

Finally, Figure 13 indicates that the effects of the tolerable duration parameter highly depend on the quality levels defined for the user. Understanding the impact of the tolerable duration parameter can help in establishing contracts with users, as well as tuning the architecture based on the performance characteristics.

IX. Related Work

Recently, TI applications have begun to provide sophisticated features such as extensive configurability, high-resolution audio and video, and haptic sensing. These systems often rely on a multitude of specialized hardware and software components. To meet the bandwidth demands of the high-resolution audio and video, these systems typically run over dedicated networking infrastructures. Accordingly, packet loss rates and delays are low, and thus have only a minor impact on the dependability of TI applications compared to individual component failures.

Existing TI applications [30], [7], [31], [32] do not consider failures in the design of their architecture. For this reason, applications developed on top of these architectures do not provide maximum QoE to the end users in the presence of failures. Furthermore, the established standards [33], [34] for evaluating the QoE of traditional teleconferencing applications are mostly concerned with network level failures, and do not sufficiently cover failure of hardware components, such as microphones, workstations, etc. Further, [17] claims that the realism offered by TI cannot be matched by teleconferencing applications, since users can also perform other activities, such as dancing. Hence, these standards are inappropriate for evaluating the QoE of TI applications.

The common way to quantify the QoE for multimedia applications is to use a subjective assessment method [17], where audience members of various ages are requested to rate the performance on a scale from 1 to 5, and then compute the mean. However, this method is ineffective as a means to improve the QoE of WO performances, as it is expensive and time consuming to conduct surveys and performances. Furthermore, it allows for ambiguous results due to the differences in expectations among the participants [35].

With respect to quantifying QoE, [36] used a pentagram modeling framework targeting VoIP services, while [37] targets network failures with a pseudo-subjective quality assessment method to quantify the QoE using a neural network. However, both frameworks are limited, and do not capture the complex characteristics of TI applications.

Just noticeable difference (JND) [38] is another concept in psychology that might be relevant in our context. It is defined as the amount of deviation that must occur for the difference to be noticeable. JND relates to magnitude of changes, while unperceivable failures relate to their duration. We do not explicitly use JND in the paper, but it could help in defining the QoE levels for users.

X. Conclusion

This paper presents a novel solution to the problem of quantifying user perceived QoE in the presence of failures in TI applications. Our approach, which is based on the concept of "perceived reliability", takes into account characteristics of human perception. To demonstrate the feasibility of our approach, we have designed and implemented our modeling framework based on Stochastic Activity Networks. Our results have provided useful insights that have aided technicians involved in WO performances.

When designing for maximum QoE, a modeling framework should allow for early design decisions by comparing different architectural variants. Our proposed *perceived reliability* concept, and the related modeling framework provides an effective method to *quantify* the QoE for users of TI applications in presence of failures. While our framework was developed for the WO system, it can be used for QoE evaluation of a wide variety of distributed multimedia applications, such as video conferencing, online gaming, or even life-critical multimedia applications, such as distributed collaborative computer-assisted surgery [39].

References

[1] T. Hampel, T. Bopp, and R. Hinn, "A peer-to-peer architecture for massive multiplayer online games," in *Proc. ACM NetGames*, 2006.

[2] "The World Opera," http://theworldopera.org, (Accessed: 09/02/2015).

[3] R. T. Kellogg, *Fundamentals of cognitive psychology*. Sage, 2011.

[4] A. Avižienis, J.-C. Laprie, B. Randel, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, pp. 11–33, 2004.

[5] W. H. Sanders and J. F. Meyer, "Stochastic activity networks: formal definitions and concepts," in *Lectures on formal methods and performance analysis*, 2002, pp. 315–343.

[6] S. Gaonkar *et al.*, "Performance and dependability modeling with Möbius," *SIGMETRICS Perform. Eval. Rev.*, vol. 36, no. 4, pp. 16–21, March 2009.

[7] R. Zimmermann, E. Chew, S. A. Ay, and M. Pawar, "Distributed musical performances: Architecture and stream management," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 4, no. 2, pp. 14:1–14:23, May 2008.

[8] Z. Yang, B. Yu, W. Wu, K. Nahrstedt, R. Diankov, and R. Bajscy, "A study of collaborative dancing in tele-immersive environments," in *Proc. IEEE Multimedia (ISM)*, 2006, pp. 177–184.

[9] A. Arefin, Z. Huang, R. Rivas, S. Shi, W. Wu, and K. Nahrstedt, "Tele-immersive gaming for everybody," in *Proc. ACM Multimedia (MM)*, 2011, pp. 783–784.

[10] A. Ståhlbröst, *Human-centric evaluation of innovation*. Luleå tekniska universitet, 2006.

[11] I. Rec, "P. 10: Vocabulary for performance and quality of service, amendment 2: New definitions for inclusion in recommendation itu-t p. 10/g. 100," *Int. Telecomm. Union, Geneva*, 2008.

[12] G. Albaum, "The likert scale revisited," *Journal-Market research society*, vol. 39, pp. 331–348, 1997.

[13] K. De Moor, "Are engineers from mars and users from venus?: bridging gaps in quality of experience research: reflections on and experiences from an interdisciplinary journey," Ph.D. dissertation, Ghent University, 2012.

[14] K.-T. Chen, C.-C. Tu, and W.-C. Xiao, "Oneclick: A framework for measuring network quality of experience," in *Proc. IEEE INFOCOM*, 2009, pp. 702–710.

[15] N. R. Veeraragavan, H. Meling, and R. Vitenberg, "QoE estimation models for tele-immersive applications," in *Proc. IEEE EUROCON*, 2013, pp. 154–161.

[16] W. Wu, A. Arefin, R. Rivas, K. Nahrstedt, R. Sheppard, and Z. Yang, "Quality of experience in distributed interactive multimedia environments: toward a theoretical framework," in *Proc. ACM Multimedia (MM)*, 2009, pp. 481–490.

[17] Z. Huang *et al.*, "Towards the understanding of human perceptual quality in tele-immersive shared activity," in *Proc. ACM Multimedia Systems (MMSys)*, 2012, pp. 29–34.

[18] M. Sithu and Y. Ishibashi, "QoE assessment of joint haptic drum performance: Effect of local lag control," in *Proc. IEEE Global Conf. on Consumer Electronics (GCCE)*, 2013, pp. 461–465.

[19] A. Olmos *et al.*, "Exploring the role of latency and orchestra placement on the networked performance of a distributed opera," in *Proc. Annual Int. Workshop on Presence*, 2009.

[20] N. R. Veeraragavan, A. Bondavalli, L. Montecchi, R. Vitenberg, N. Nostro, and H. Meling, "Understanding the quality of experience in modern distributed interactive multimedia applications in presence of failures: Metrics and analysis," in *Proc. Annual ACM Symp. on Applied Computing (SAC)*, 2013, pp. 439–446.

[21] A. Bondavalli, P. Lollini, and L. Montecchi, "QoS Perceived by Users of Ubiquitous UMTS: Compositional Models and Thorough Analysis," *Journal of Software*, vol. 4, no. 7, pp. 675–685, September 2009.

[22] L. Montecchi, P. Lollini, and A. Bondavalli, "A DSL-Supported Workflow for the Automated Assembly of Large Performability Models," in *Proc. European Depend. Comput. Conf. (EDCC)*, 2014, pp. 82–93.

[23] G. Ciardo, R. German, and C. Lindemann, "A characterization of the stochastic process underlying a stochastic petri net," in *Workshop on Petri Nets and Performance Models*, 1993, pp. 170–179.

[24] W. H. Sanders and J. F. Meyer, "Reduced base model construction methods for stochastic activity networks," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 1, pp. 25–36, January 1991.

[25] L. Montecchi, N. Nostro, N. R. Veeraragavan, R. Vitenberg, H. Meling, and A. Bondavalli, "Stochastic activity networks model for the evaluation of the world opera system," http://rcl.dsi.unifi.it/publication/show/719-2, October 2015, v2.1, (Accessed: 14/10/2015).

[26] S. Rosen, "Temporal information in speech: acoustic, auditory and linguistic aspects," *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, vol. 336, no. 1278, pp. 367–373, 1992.

[27] D. M. Eagleman and T. J. Sejnowski, "Motion integration and postdiction in visual awareness," *Science*, vol. 287, no. 5460, pp. 2036–2038, 2000.

[28] K. S. Trivedi, *Probability and Statistics with Reliability, Queueing, and Computer Science Applications, 2nd Edition*. Wiley-Interscience, October 2001.

[29] K. S. Trivedi, R. Vasireddy, D. Trindale, S. Nathan, and R. Castro, "Modeling high availability," in *Proc. IEEE Pacific Rim Int. Symp. on Dependable Computing (PRDC)*, 2006, pp. 154–164.

[30] W. Wu, Z. Yang, D. Jin, and K. Nahrstedt, "Implementing a distributed tele-immersive system," in *Proc. IEEE Multimedia (ISM)*, 2008, pp. 477–484.

[31] A. Hamam, A. E. Saddik, and J. Alja'am, "A quality of experience model for haptic virtual environments," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 10, no. 3, pp. 28:1–28:23, Apr. 2014.

[32] P. Xia and K. Nahrstedt, "Object-level bandwidth adaptation framework for 3D tele-immersive system," in *Proc. Int. Conf. on Multimedia and Expo (ICME)*, 2013, pp. 1–6.

[33] "ITU-G.1070. Opinion model for video-telephony applications," 2007.

[34] "ITU-G.107. The E-model, a computational model for use in transmission planning," 2008.

[35] H. Knoche, H. G. De Meer, and D. Kirsh, "Utility curves: Mean opinion scores considered biased," in *Proc. Workshop on Quality of Service*, 1999, pp. 12–14.

[36] Y. Gong, F. Yang, L. Huang, and S. Su, "Model-based approach to measuring quality of experience," in *Proc. Int. Conf. on Emerging Network Intelligence*, 2009, pp. 29–32.

[37] A. P. C. da Silva, M. Varela, E. de Souza e Silva, R. M. M. Leão, and G. Rubino, "Quality assessment of interactive voice applications," *Comput. Netw.*, vol. 52, no. 6, pp. 1179–1192, Apr. 2008.

[38] E. Zwicker and H. Fastl, *Psychoacoustics: Facts and Models*, ser. Springer Series in Information Sciences. Springer-Verlag, 1990.

[39] H. Lufei, W. Shi, and V. Chaudhary, "M-CASEngine: a collaborative environment for computer-assisted surgery," *Int. Journal of Computer Assisted Radiology and Surgery*, vol. 1 SUPP/1, pp. 447–448, 2006.